3. Linear Programming

Note for the reader

Paragraphs from 3.1 to 3.4 are more summarised than later paragraphs, but still contain all the useful information about the chapter.
The content from these paragraphs is taken directly from the document 3a. Notes on LP.

3.1. Equivalent forms

A problem is in **standard form** if it:

1. **Only has equality constraints**, no inequality constraints are allowed
2. **Only has non-negative variables**, nil and negative variables are not allowed

To transform constraints:

- If the constraint is in the form $\underline{a}^T\underline{x} \leq \underline{b}$, I can add a **slack** variable $s \geq 0$ to let the left side of the equation reach the value of the right side, and obtain:

$$\begin{cases} \underline{a}^T\underline{x} + s = \underline{b} \\ s \geq 0 \end{cases}$$

- If the constraint is in the form $\underline{a}^T\underline{x} \geq \underline{b}$, I can subtract a **surplus** variable $s \geq 0$ to let the left side of the equation reach the value of the right side, and obtain:

$$\begin{cases} \underline{a}^T\underline{x} - s = \underline{b} \\ s \geq 0 \end{cases}$$

To transform variables:

- If a variable x_j is **unrestricted in sign**, it can be expressed as the difference of its positive and negative parts:

$$\begin{cases} x_j = x_j^+ - x_j^- \\ x_j^+, x_j^- \geq 0 \end{cases}$$

After the substitution, we can delete x_j from the problem.

All other transformations are very straightforward and are related to changing the signs of constraint equations and inequalities.

3.2. Geometry of linear programming

An **unbounded feasible direction** is a vector that, placed at any point within a polyhedron P, can continue forever.

That is, the direction in which the vector is pointing is **always contained within the polyhedron**, infinitely, **regardless of the vector's length or point of origin**.

This definition can help us define polyhedra in a new way:

> **Weyl-Minkowski Theorem**
>
> Every point x of a polyhedron P can be expressed as a **convex combination of its vertices** $\underline{x}_1, \dots, \underline{x}_k$ and, if needed, an **unbounded feasible direction** $\underline{d}$ of P:
>
> $$\underline{x} = \alpha_1 \underline{x}_1 + \ldots + \alpha_k \underline{x}_k + \underline{d}, \text{ with } \alpha_i \geq 0, \ \sum_{i=1}^{k} \alpha_i = 1$$

Unbounded feasible directions are not needed in **polytopes**, which are polyhedra in which the only unbounded feasible direction is $\underline{d} = \underline{0}$. This means that polytopes can be described through just the convex combination of their vertices. For example:

$$\underline{x} = \alpha_1 \underline{x}_1 + \alpha_2 \underline{x}_2 + \alpha_3 \underline{x}_3, \text{ with } \alpha_i \geq 0 \text{ and } \sum_{i=1}^{3} \alpha_i = 1$$

Given the above information, we can enunciate the Fundamental Theorem of Linear Programming:

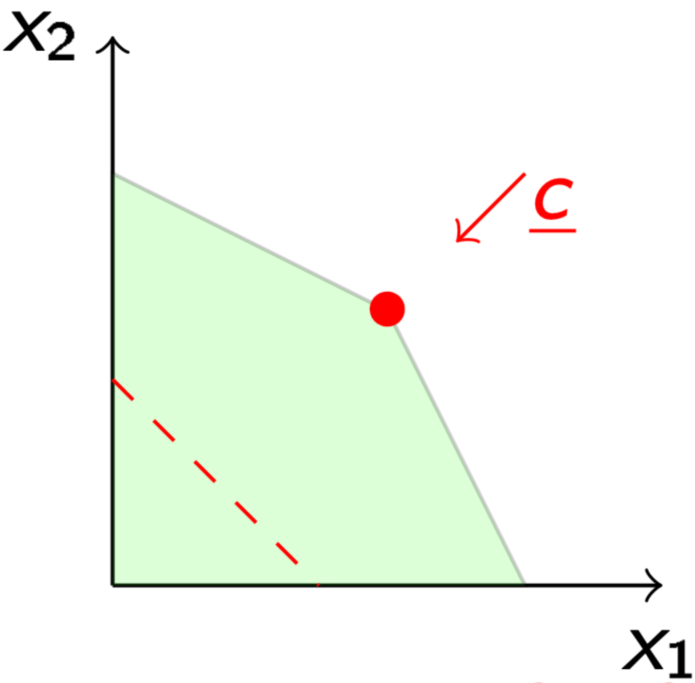
> **Fundamental Theorem of Linear Programming**
>
> Consider a Linear Programming problem such as:

$$\min\ \{\underline{c}^T\underline{x} : \underline{x} \in P\}$$

Where $P \subseteq \mathbb{R}^n$ is a non-empty polyhedron in standard or canonical form. Then, either there exists at least one **optimal vertex**, or the value of the objective function is **unbounded below** on P.

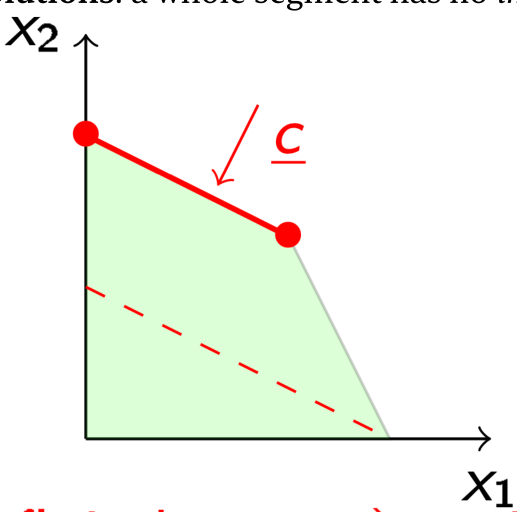
There are four types of Linear Programming problems:

1. **Unique optimal solution**: only one vertex has no *improving directions*:



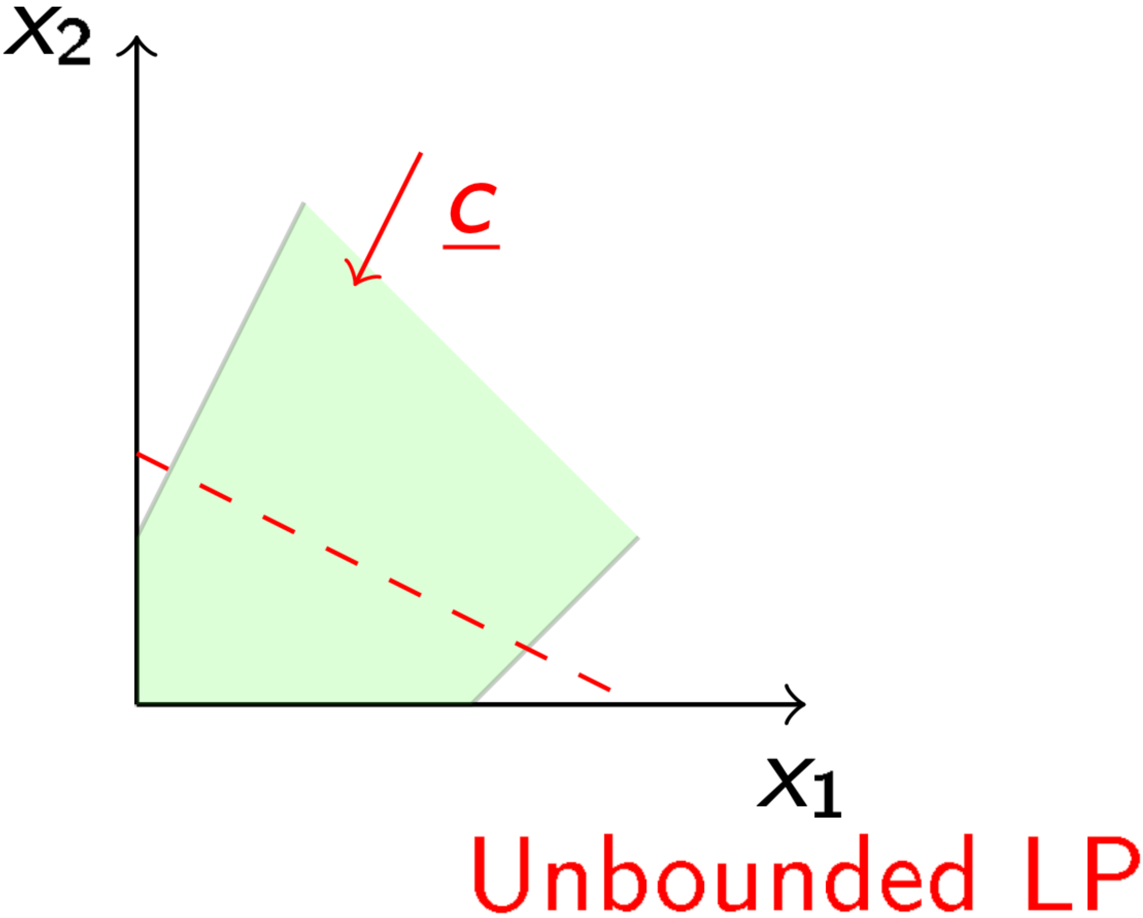
A unique optimal solution

2. **Multiple optimal solutions**: a whole segment has no *improving directions*:

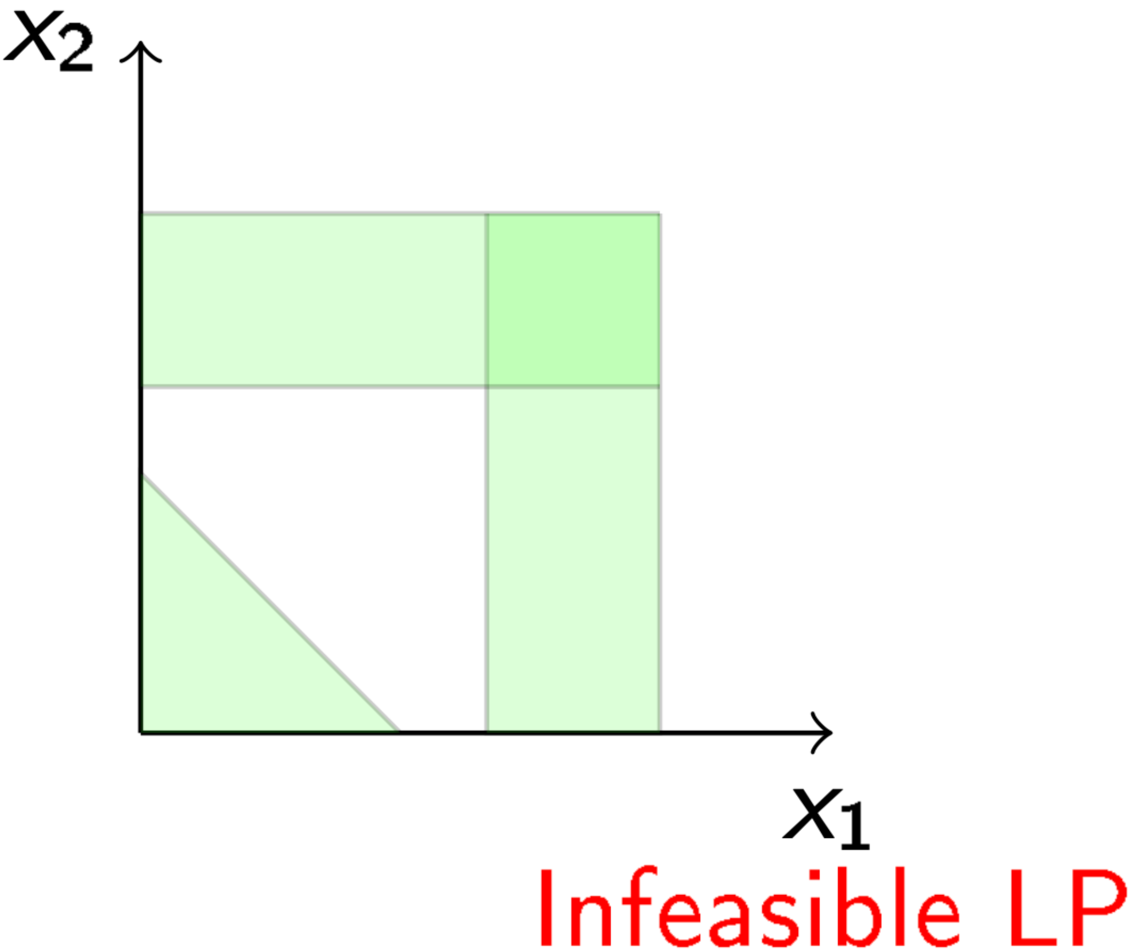


Multiple (infinitely many) optimal solutions

3. **Unbounded solution**: no optimal vertex can be found:



4. **Infeasible solution**: the feasible region is an empty polyhedron:



3.3. Basic feasible solutions and vertices of polyhedra

Since we discovered that we can find optimal solutions by checking the vertices of feasible polyhedra, let's find a way to compute vertices algebraically.

We know that vertices correspond to the intersections of hyperplanes associated to the n constraint inequalities in Linear Programming problems. However, in standard form, we have no inequalities.

In order to solve this problem, we can begin from the canonical form of a problem and transform it into standard form in the ways we've seen before.

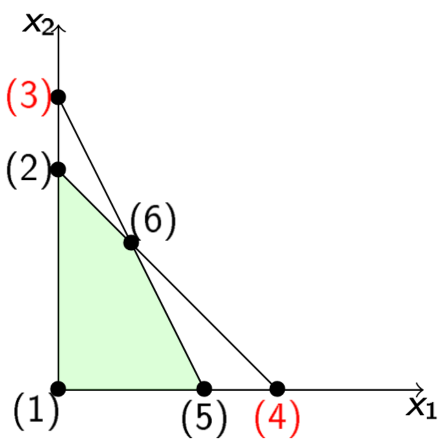
If this is done, every constraint in P now **corresponds to a slack variable** in the new polyhedron P' and, when $s_i = 0$, the **constraint is satisfied** through an equality.

It must be noted that, when computing intersections in P', if any variable or slack variable is negative, the found solution is infeasible:

Example

Compute all the intersections

$$\begin{aligned} x_1 + x_2 + s_1 &= 6 \\ 2x_1 + x_2 + s_2 &= 8 \\ x_1, x_2, s_1, s_2 &\geq 0 \end{aligned}$$



(1) $x_1 = 0, x_2 = 0 \Rightarrow s_1 = 6, s_2 = 8$

(2) $x_1 = 0, s_1 = 0 \Rightarrow x_2 = 6, s_2 = 2$

(3) $x_1 = 0, s_2 = 0 \Rightarrow x_2 = 8, s_1 = -2$

(4) $x_2 = 0, s_1 = 0 \Rightarrow x_1 = 6, s_2 = -4$

(5) $x_2 = 0, s_2 = 0 \Rightarrow x_1 = 4, s_1 = 2$

(6) $s_1 = 0, s_2 = 0 \Rightarrow x_1 = 2, x_2 = 4$

We must now define which are the vertices of a polyhedron in standard form. It is quite simple:

> **Property of standard form polyhedra**
>
> For any polyhedron $P = \{\underline{x} \in \mathbb{R}^n : A\underline{x} = \underline{b},\ \underline{x} \geq \underline{0}\}$:
>
> 1. The **facets** are obtained by setting one variable to zero
> 2. The **vertices** are obtained by setting $n - m$ variables to zero, where:
>   - n is the number of variables in the problem
>   - m is the number of constraints in the problem

We can now formally characterise vertices.

Consider any polyhedron in standard form and assume that $A \in \mathbb{R}^{m \times n}$ is such that A is of rank m. This is **equivalent to assume** that there are **no redundant constraints** (so all rows and columns in the matrix are linearly independent).

Now:

- If $m = n$, there is a unique solution to $A\underline{x} = \underline{b}$, which is:

$$\underline{x} = A^{-1}\underline{b}$$

- If $m < n$, there are ∞^{n-m} solutions of $A\underline{x} = \underline{b}$. We can also say that the system has $n - m$ degrees of freedom or that $n - m$ variables can be fixed arbitrarily. If we do fix $m - n$ solutions arbitrarily, we **find a vertex**.

If we consider the A matrix from before, we can define a **basis** of such matrix as a subset of m columns of A that are linearly independent and form an $m \times m$ **non-singular** matrix B. We can so redefine A as:

$$A = [\underbrace{B}_{m} | \underbrace{N}_{n-m}]$$

Now, let $\underline{x} = [\underline{x}_B^T | \underline{x}_N^T]^T$.

Then, any system $A\underline{x} = \underline{b}$ can be written as:

$$B\underline{x}_B + N\underline{x}_N = \underline{b}$$

For any set of values for $\underline{x}_N$, if B is non-singular, we have:

$$\underline{x}_B = B^{-1}b - B^{-1}N\underline{x}_N$$

Finally we can say that:

- A **basic solution** is a solution obtained by setting $\underline{x}_N = \underline{0}$ and consequently letting $\underline{x}_B = B^{-1}\underline{b}$.
- A basic solution with $\underline{x}_B \geq 0$ is known as a **basic feasible solution**.
- We call the variables in $\underline{x}_B$ **basic variables** and the variables in $\underline{x}_N$ **non-basic variables**.

Now that we defined everything that we need, we can enunciate the following theorem:

Theorem

$\underline{x} \in \mathbb{R}^n$ is a **basic feasible solution** if and only if $\underline{x}$ is a **vertex** of $P = \{\underline{x} \in \mathbb{R}^n : A\underline{x} = \underline{b},\ \underline{x} \geq \underline{0}\}$.

The number of basic feasible solutions of any given Linear Programming problem is always smaller or equal to $\binom{n}{m}$.

3.4. The Simplex method

3.4.1. Optimality test

Given a Linear Programming problem like $\min\ \{\underline{c}^T\underline{x} : A\underline{x} = \underline{b},\ \underline{x} > \underline{0}\}$ and a feasible basis B of A, $A\underline{x} = \underline{b}$ can be rewritten as:

$$B\underline{x}_B + N\underline{x}_N = \underline{b}$$

From this equation we can extract $\underline{x}_B$ as:

$$\underline{x}_B = B^{-1}\underline{b} - B^{-1}N\underline{x}_N, \text{ with } B^{-1}\underline{b} \geq 0$$

Then, a **basic feasible solution** is such that $\underline{x}_B = B^{-1}\underline{b},\ \underline{x}_N = \underline{0}$, as we said earlier.

By substitution, we can express the objective function in terms of the non-basic variables only:

$$\begin{aligned}\underline{c}^T\underline{x} &= \left(\underline{c}_B^T \mid \underline{c}_N^T\right)\begin{bmatrix}\underline{x}_B \\ \underline{x}_N\end{bmatrix} \\ &= \left(\underline{c}_B^T \mid \underline{c}_N^T\right)\begin{bmatrix}B^{-1}\underline{b} - B^{-1}N\underline{x}_N \\ \underline{x}_N\end{bmatrix} \\ &= \underline{c}_B^T B^{-1}\underline{b} + \left(\underline{c}_N^T - \underline{c}_B^T B^{-1}N\right)\underline{x}_N \\ &= z_0 + \underline{\overline{c}}_N^T\underline{x}_N\end{aligned}$$

Now we can say:

- The parameter $z_0 = \underline{c}_B^T B^{-1}\underline{b}$ is the cost of $\underline{x} = \left(\underline{x}_B \mid \underline{x}_N\right)^T$, where $\underline{x}$ is a basic feasible solution
- The parameter $\underline{\overline{c}}_N^T = \underline{c}_N^T - \underline{c}_B^T B^{-1}N$ defines the **reduced costs** of the non-basic variables

We can use the parameters above to define the vector of reduced costs for a basis:

Vector of reduced costs with respect to a basis B

$$\begin{aligned}\underline{\overline{c}}^T &= \underline{c}^T - \underline{c}_B^T B^{-1}A \\ &= \left[\underline{c}_B^T - \underline{c}_B^T B^{-1}B \mid \underline{c}_N^T - \underline{c}_B^T B^{-1}N\right] \\ &= \left[\underline{0}^T \mid \underline{\overline{c}}_N^T\right]\end{aligned}$$

If we consider the same Linear Programming problem as before, we can now say that:

Proposition 1

If $\underline{\overline{c}}_N \geq 0$[1], then the basic feasible solution $\left(\underline{x}_B^T \mid \underline{x}_N^T\right)$ of cost $\underline{c}_B^T B^{-1}\underline{b}$, where $\underline{x}_B = B^{-1}\underline{b} \geq 0$ and $\underline{x}_N = \underline{0}$, is a **global optimum**.

Example

Let's take a look at the following example:

$$\begin{array}{llllllll}
\min & z = & -x_1 & -x_2 & & & \\
\text{s.t.} & & x_1 & -x_2 & +s_1 & & = 1 \\
& x_1 & +x_2 & & +s_2 & & = 3 \\
& x_1, & x_2, & s_1, & s_2 & & \geq 0
\end{array}$$

If we consider a BFS with $\underline{x}_B = \left(x_1 \mid s_2\right)^T$, then we must put the other non-basic variables to zero, which means that $\underline{x}_N = \left(x_2 \mid s_1\right)^T = \underline{0}^T$. We obtain the following model:

$$\begin{array}{lllllll}
\min & z = & -x_1 & -0 & & \\
\text{s.t.} & & x_1 & -0 & +0 & = 1 \\
& x_1 & +0 & & +s_2 & = 3 \\
& x_1, & x_2, & s_1, & s_2 & \geq 0
\end{array}$$

From the above equations we can now extract the values of x_1, s_2 and z:

$$\begin{array}{lllllll}
\min & z = & -x_1 & & & = -1 \\
\text{s.t.} & & x_1 & & & = 1 \\
& & & & s_2 & = 2 \\
& x_1, & x_2, & s_1, & s_2 & \geq 0
\end{array}$$

We can also write the B and N matrices:

$$B = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, N = \begin{bmatrix} -1 & 1 \\ 1 & 0 \end{bmatrix}$$

Returning to the beginning for a minute, we note that the cost vector $\underline{c}$ is:

$$\underline{c} = \left[-1 \ -1\ 0\ 0\right]^T$$

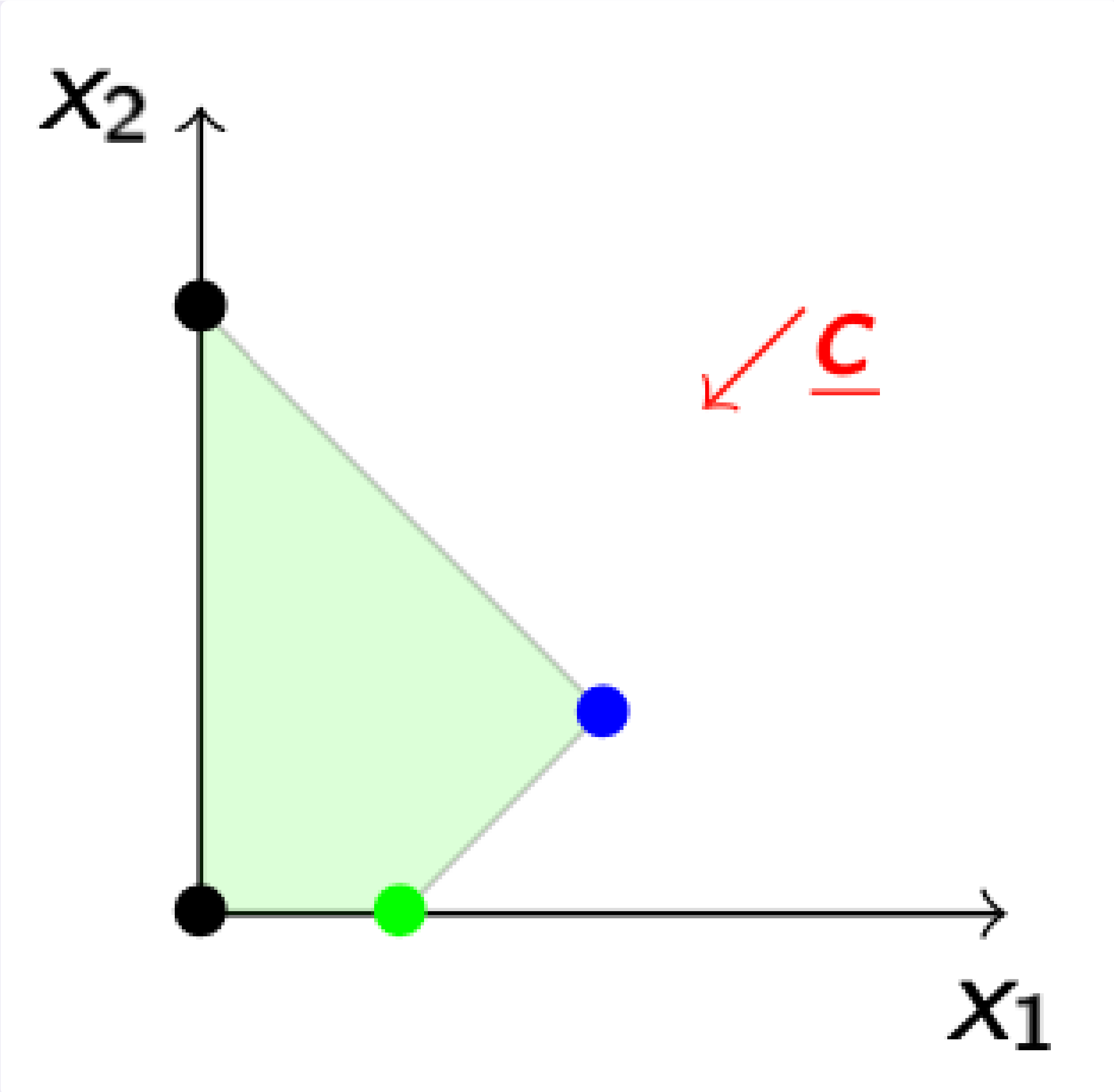
So now we can extract $\underline{c}_B^T$ and $\underline{c}_N^T$:

$$\underline{c}_B^T = \underline{c}_N^T = \left[-1\ 0\right]^T$$

Once this is done, we can calculate the reduced costs for x_2 and s_1:

$$\underline{\bar{c}}_N^T = \underline{c}_N^T - \underline{c}_B^T B^{-1} N = \left[-2\ 0\right]^T$$

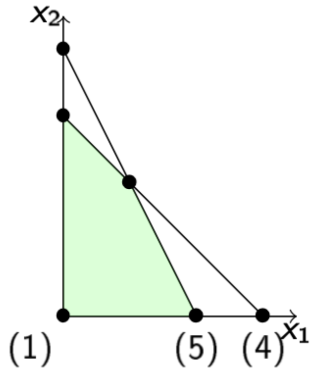
Since we know that we are in a global optimum if $\underline{\bar{c}}_N^T \geq 0$, we see that we can move to a better vertex thanks to the reduced cost vector. Specifically, if x_2 can be increased from 0 to 1 while keeping $s_1 = 0$, then z can vary by -2. In practice, we now know that, in order to minimise z more, we can move from the green vertex $(1, 0)$ to the blue vertex $(2, 1)$, as shown below:



3.4.2. Move to an adjacent vertex

When moving from a vertex to another, we are simply substituting a column from B with a column from N. To better understand this, take a look at the example below:

Example



$$\begin{aligned} x_1 + x_2 + s_1 \quad &= 6 \\ 2x_1 + x_2 \quad + s_2 &= 8 \\ x_1,\ x_2,\ s_1,\ s_2 &\geq 0 \end{aligned}$$

Move from vertex (1) to vertex (5):

- In (1) $x_1 = 0, x_2 = 0 \Rightarrow s_1 = 6, s_2 = 8$ with $\underline{x}_B = (s_1, s_2)^T$, $\underline{x}_N = (x_1, x_2)T$
- In (5) $x_2 = 0, s_2 = 0 \Rightarrow x_1 = 4, s_1 = 2$ with $\underline{x}_B = (x_1, s_1)^T$, $\underline{x}_N = (x_2, s_2)^T$

Thus, x_1 enters the basis B and s_2 exits the basis B.

In order to understand *where* to move, we can express the basic variables in terms of the non-basic ones and "choose" a direction to move in. Taking the same example as before, we can write:

$$s_1 = 6 - x_1 - x_2$$
$$s_2 = 8 - 2x_1 - x_2$$

If we decide to move on x_1, then we set $x_2 = 0$ and increase x_1. But how do we know how much we should increase it by?

Taking the previous equations, we can say that, since $x_1, x_2, s_1, s_2 \geq 0$:

$$s_1 = 6 - x_1 \geq 0$$
$$s_2 = 8 - 2x_1 \geq 0$$

Solving these inequalities gives us two upper bounds:

$$x_1 \leq 6$$
$$x_1 \leq 4$$

We can now check which solutions are feasible by substituting their value in the equations above:

- For $x_1 = 6$:

  $$s_1 = 6 - 6 = 0$$
  $$s_2 = 8 - 12 = -4$$

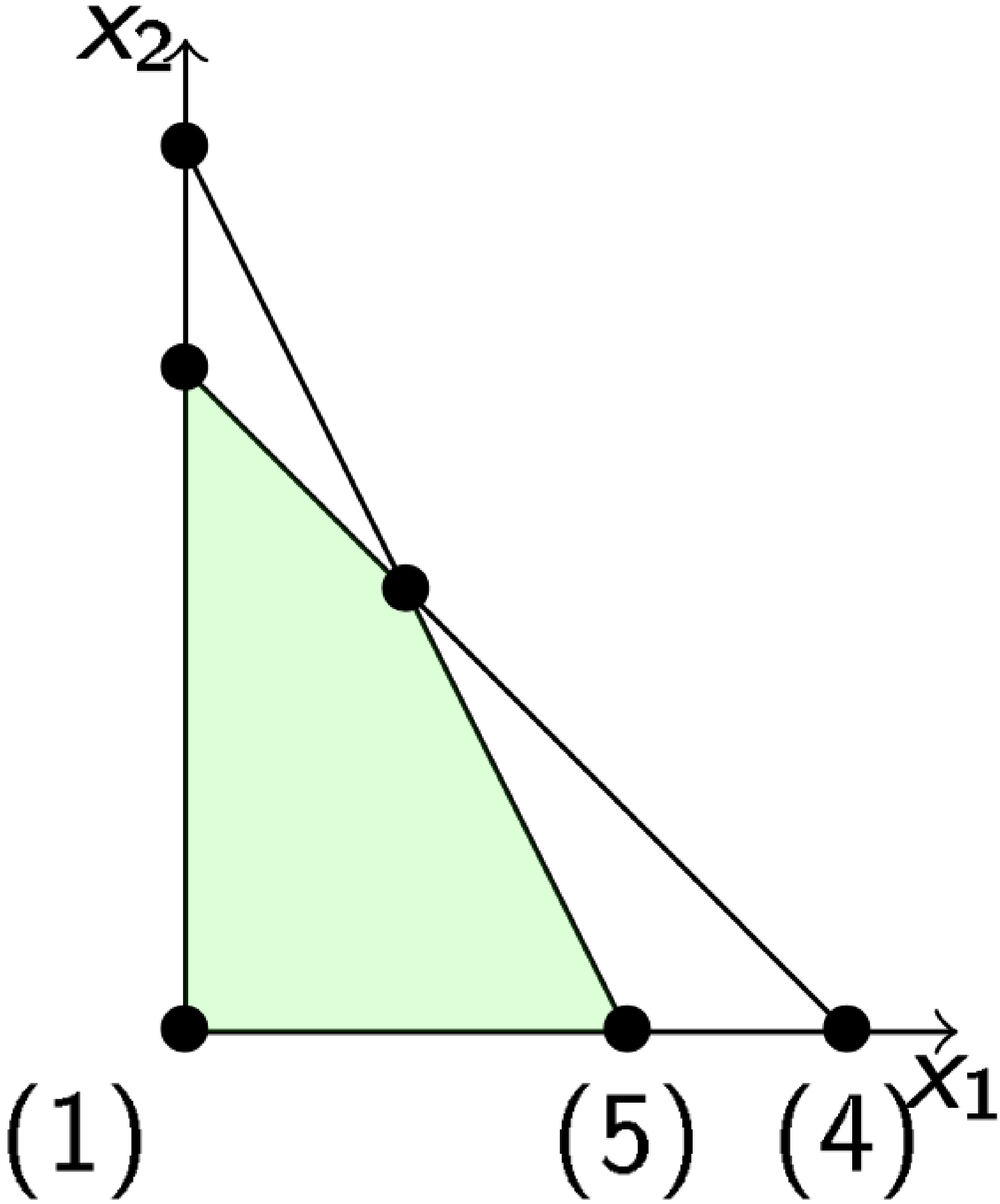
  Since $s_2 < 0$, the solution is infeasible.

- For $x_1 = 4$:

$$s_1 = 6 - 4 = 2$$
$$s_2 = 8 - 8 = 0$$

  Since both s_1 and s_2 are non-negative, the solution is feasible.

We can also see the result above through the graph (where vertex 5 is the feasible solution, while vertex 4 is the infeasible one):



In general, we can apply the **change of basis** method to minimise a Linear Programming problem:

Change of basis for a minimisation Linear Programming problem

Let B be a feasible basis and x_s in $\underline{x}_N$ a non-basic variable with a reduced cost $\overline{c}_s < 0$. Now:

1. **Increase x_s as much as possible** (we say that x_s *enters the basis*) while keeping the other non-basic variables equal to zero
2. The basic variable x_r in $\underline{x}_B$ such that $x_r \geq 0$ imposes the **tightest upper bound** θ^* on the increase of x_s (we say that x_r *leaves the basis*)
3. If $\theta^* > 0$, the new basic feasible solution has a better objective function value. The new basis differs by a single column with respect to the previous one.

Since the basis B changes when moving from a canonical form to the next, one may think that B^{-1} should be computed all over again. However, we only need a **pivoting** operation in order to recompute it:

Pivoting operation

Given the system $A\underline{x} = \underline{b}$:

1. Select a coefficient $\overline{a}_{rs} \neq 0$: this will be our **pivot**
2. Divide the r-th row by $\overline{a}_{rs}$
3. For each row i with $i \neq r$ and $\overline{a}_{is} \neq 0$, subtract the resulting r-th row multiplied by $\overline{a}_{is}$.

This is similar to the operations used in the Gaussian elimination method to solve systems of linear equations.

Let's look at an example starting from the following matrices:

$$A = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & 1 & 0 & 1 \end{bmatrix}, \underline{b} = \begin{bmatrix} 6 \\ 8 \end{bmatrix}$$

We will now write the two matrices next to each other, highlighting the chosen row and column for our example:

$$
\begin{array}{cccccc}
 & s & & & & \\
 & 1 & 1 & 1 & 0 & 6 \\
r & 2 & 1 & 0 & 1 & 8
\end{array}
$$

Now, let's apply the algorithm as written above: we chose our 2 for a pivot, so now we divide the row containing 2, which is r, by 2:

$$
\begin{array}{cccccc}
 & s & & & & \\
 & 1 & 1 & 1 & 0 & 6 \\
r & 1 & \frac{1}{2} & 0 & \frac{1}{2} & 4
\end{array}
$$

Finally, we subtract each row in the matrix that is not r with the resulting r row, multiplied by the coefficient in the same column as 2. In this case, we only have another row, in which $\bar{a}_{is} = 1$:

$$
\begin{array}{cccccc}
 & s & & & & \\
i & 0 & \frac{1}{2} & 1 & -\frac{1}{2} & 2 \\
r & 1 & \frac{1}{2} & 0 & \frac{1}{2} & 4
\end{array}
$$

Summing up, in order to move to a better vertex, we must answer the following two questions:

1. Which non-basic variable enters the basis?
   - Any variable with reduced cost $\bar{c}_j < 0$
   - A variable that yields the maximum Δz
2. Which basic variable leaves the basis?
   - This is decided via the **min ratio test**: the variable of index i with smallest $\frac{\bar{b}_i}{\bar{a}_{is}} = \theta^*$ among those with $\bar{a}_{is} > 0$ is the one that leaves the basis

Both variables can be decided by applying **Bland's rule**:

Bland's rule

The variable that enters the basis is the one with index s such that:

$$s = \min\ \{j : \bar{c}_j < 0\}$$

The variable that leaves the basis is the one with index r such that:

$$r = \min\ \left\{i : \frac{\bar{b}_i}{\bar{a}_{is}} = \theta^*,\ \bar{a}_{is} > 0\right\}$$

A Linear Programming problem is said to be **unbound** if there exists a variable with a reduced cost $\overline{c}_j < 0$ with $\overline{a}_{ij} \leq 0,\ \forall i$. In other words, a problem is unbounded if **no element** of the j-th column **can be a pivot**.

3.4.3. Tableau representation

Linear Programming problems in standard form can easily be visualised in the so called **Tableau representation**.

Let's take the following problem as an example:

$$
\begin{array}{llllll}
\min & -x_1 & -x_2 & & & \\
\text{s.t.} & 6x_1 & +4x_2 & +x_3 & & = 24 \\
 & 3x_1 & -2x_2 & & +x_4 & = 6 \\
 & x_1, & x_2, & x_3, & x_4 & \geq 0
\end{array}
$$

The Tableau representation of the above problem is:

$$
\begin{array}{cccccc}
 & & x_1 & x_2 & x_3 & x_4 \\
-z & 0 & -1 & -1 & 0 & 0 \\
x_3 & 24 & 6 & 4 & 1 & 0 \\
x_4 & 6 & 3 & -2 & 0 & 1
\end{array}
$$

Applying Bland's rule, we find that the variable entering the basis is x_1, while the one exiting the basis is x_4. We find that the pivot is 3, as shown:

$$
\begin{array}{cccccc}
 & & \color{blue}{x_1} & x_2 & x_3 & x_4 \\
-z & 0 & -1 & -1 & 0 & 0 \\
x_3 & 24 & 6 & 4 & 1 & 0 \\
\color{blue}{x_4} & 6 & \color{red}{3} & -2 & 0 & 1
\end{array}
$$

Now, we can divide the x_4 row by the pivot and subtract it from the x_3 and $-z$ rows:

$$
\begin{array}{cccccc}
 & & x_1 & x_2 & x_3 & x_4 \\
-z & 2 & 0 & -\frac{5}{3} & 0 & \frac{1}{3} \\
x_3 & 12 & 0 & 8 & 1 & -2 \\
\color{red}{x_1} & 2 & 1 & -\frac{2}{3} & 0 & \frac{1}{3}
\end{array}
$$

We can continue by applying iteratively the above operations, until we only have non-negative reduced costs:

$$\begin{array}{cccccc} & & x_1 & x_2 & x_3 & x_4 \\ -z & 6 & \frac{1}{2} & 0 & \frac{1}{4} & 0 \\ x_2 & 6 & \frac{3}{2} & 1 & \frac{1}{4} & 0 \\ x_4 & 18 & 6 & 0 & \frac{1}{2} & 1 \end{array}$$

So we can conclude that the basic feasible solution (composed of nil non-basic variables and non-negative basic variables) is $\underline{x}^* = (0, 6, 0, 18)^T$.

3.4.4. Degenerate basic feasible solutions

In some cases, basic variables **can be nil**. If that happens, the basic feasible solution found is said to be **degenerate**.

A solution with more than $n - m$ zeroes (where n is the number of variables and m is the number of constraints, as mentioned before) corresponds to more than one basis, which also means that more than n constraints are satisfied with equality.

In the presence of degenerate basic feasible solutions, a basis change **may not decrease the objective function value**, since θ^* could be zero and the new optimised solution remains the same as the previous one.

It should be noted that a degenerate BFS can arise from a non-degenerate one: even if $\theta^* > 0$, several basic variables may become nil when x_s is increased to θ^*.

The main problem with degenerate bases is that one can **cycle through a sequence** of them when solving a problem. Several anti-cycling rules have been proposed for the choice of the variables that enter and exit the bases, like Bland's rule.

When applying Bland's rule, the Simplex algorithm is guaranteed to terminate after at most $\binom{n}{m}$ iterations, since the number of pivots is finite.

In some pathological cases, the number of iterations may grow exponentially with respect to n or m. However, the Simplex algorithm is overall very efficient. Extensive experimental campaigns show that the number of iterations grows linearly with respect to m and very slowly with respect to n.

3.4.5. Two-phase Simplex method

Let's take the following problem as an example:

$$\begin{array}{llllll}
\min & x_1 & & +x_3 & \\
\text{s.t.} & x_1 & +2x_2 & & \leq 5 \\
& & x_2 & +2x_3 & = 6 \\
& x_1, & x_2, & x_3 & \geq 0
\end{array}$$

Let's transform it into standard form:

$$\begin{array}{lllllll}
\min & x_1 & & +x_3 & & \\
\text{s.t.} & x_1 & +2x_2 & & +x_4 & = 5 \\
& & x_2 & +2x_3 & & = 6 \\
& x_1, & x_2, & x_3, & x_4 & \geq 0
\end{array}$$

We notice that the problem is not in canonical form: that is, the A matrix does not contain a 2×2 submatrix which is the identity matrix.

In order to be able to solve this problem, we can use an **auxiliary, artificial problem** which has this form:

$$\begin{array}{ll}
\min & v = \sum_{i=1}^{m} y_i \\
\text{s.t.} & A\underline{x} + I\underline{y} = \underline{b} \\
& \underline{x}, \underline{y} \geq 0
\end{array}$$

Basically, we are adding some **artificial variables** that can help us find an identity matrix in the original problem (which we'll identify as P) by **minimising the sum of the artificial variables** in the auxiliary problem (which we'll identify as P_A).

One obvious initial basic solution for P_A is:

$$\begin{cases} \underline{y} = \underline{b} \geq 0 \\ \underline{x} = \underline{0} \end{cases}$$

Now, we have two options:

1. If $v^* > 0$, then P is **infeasible**
2. If $v^* = 0$, then clearly $\underline{y}^* = 0$ and $\underline{x}^*$ is a basic feasible solution of P

Depending on whether or not the auxiliary variables y_i are basic, P_A is solved differently:

- If y_i are non-basic $\forall i$, then the corresponding columns are deleted and a Tableau representation is obtained with respect to some basis. The objective function row (the z row) must be determined by substitution.

- If at least one y_i variable is basic (which also means the basic feasible solution is degenerate), a pivot operation is performed with respect to a non-zero coefficient of the row of y_i so as to exchange y_i with any other non-basic variable x_j.

Let's look at an example: our main problem P is:

$$\begin{array}{lrrrl} \min & x_1 & +x_2 & +10x_3 & \\ \text{s.t.} & & x_2 & +4x_3 & = 2 \\ & -2x_1 & +x_2 & -6x_3 & = 2 \\ & x_1, & x_2, & x_3 & \geq 0 \end{array}$$

We define our auxiliary problem P_A as:

$$\begin{array}{lrrrrrl} \min & v & & & = y_1 & +y_2 & \\ \text{s.t.} & & x_2 & +4x_3 & +y_1 & & = 2 \\ & -2x_1 & +x_2 & -6x_3 & & +y_2 & = 2 \\ & x_1, & x_2, & x_3, & y_1, & y_2 & \geq 0 \end{array}$$

We can write $v = y_1 + y_2$ in canonical form by expressing y_1 and y_2 in terms of x_1, x_2 and x_3. Now we obtain the Tableau form like so:

$$\begin{array}{cccccccc} & & x_1 & x_2 & x_3 & y_1 & y_2 \\ -v & -4 & 2 & -2 & 2 & 0 & 0 \\ y_1 & 2 & 0 & 1 & 4 & 1 & 0 \\ y_2 & 2 & -2 & 1 & -6 & 0 & 1 \end{array}$$

From here on, we can apply the Simplex method as we've seen earlier, and we reach the end of the first phase of the two-phase Simplex method by obtaining the optimal solution for P_A:

$$\begin{cases} \underline{x}^* = (0, 2, 0)^T \\ \underline{y}^* = (0, 0)^T \end{cases}$$

Now, we need to express the auxiliary problem in a basis consisting of the x_i variables only, to return to the original problem. We obtain:

$$\begin{array}{ccccccc} & & x_1 & x_2 & x_3 & y_1 & y_2 \\ -v & 0 & 0 & 0 & 0 & 1 & 1 \\ x_2 & 2 & 0 & 1 & 4 & 1 & 0 \\ x_1 & 0 & 1 & 0 & 5 & \frac{1}{2} & -\frac{1}{2} \end{array}$$

We note that the optimal solution $\underline{x}^*$ found earlier is also a basic feasible solution of P.

The only thing we're missing now is an expression of the original objective function in only non-basic variables. The objective function of P was:

$$z = x_1 + x_2 + 10x_3$$

Looking at the Tableau representation, we obtain the following equations:

$$\begin{cases} 2 = x_2 + 4x_3 \\ 0 = x_1 + 5x_3 \end{cases}$$

By substituting in the objective function, we obtain:

$$z = 2 + x_3$$

We can now write the Tableau representation for P and enter phase two of the two-phase Simplex method, which consists in optimising the original problem:

$$\begin{array}{ccccc} & & x_1 & x_2 & x_3 \\ -z & 2 & 0 & 0 & 1 \\ x_2 & 2 & 0 & 1 & 4 \\ x_1 & 0 & 1 & 0 & 5 \end{array}$$

Fortunately, we can see that the solution we found earlier is already optimal (there are no negative costs), so there is no need to complete phase two.

Note for the reader

The following paragraphs are more detailed than the ones above.

3.5. Linear Programming duality

Given any minimisation problem, we can associate a related maximisation problem to it. This, of course, works the other way around too.

The two problems will likely have different spaces and objective functions. However, generally, their optimal value will coincide.

The fact that we can formulate a dual problem to the one we currently have is useful, especially when we have to estimate the optimal value of our main problem.

Let's take a look at this example:

$$\begin{array}{llllllll}
\max & z & = 4x_1 & +x_2 & +5x_3 & +3x_4 & & \\
\text{s.t.} & & x_1 & -x_2 & -x_3 & +3x_4 & \leq 1 & (1) \\
& & 5x_1 & +x_2 & +3x_3 & +8x_4 & \leq 55 & (2) \\
& & -x_1 & +2x_2 & +3x_3 & -5x_4 & \leq 3 & (3) \\
& & x_1, & x_2, & x_3, & x_4 & \geq 0 &
\end{array}$$

The feasible solutions of this problem provide us with lower bounds for z^*:

- $(0,0,1,0) \implies z^* \geq 5$
- $(2,1,1,1/3) \implies z^* \geq 15$
- $(3,0,2,0) \implies z^* \geq 22$

Unfortunately, we can't be sure about which one is the best lower bound. However, by multiplying constraint 2 by $5/3$, we obtain an inequality that dominates the objective function:

$$\begin{aligned}
\frac{25}{3}x_1 + \frac{5}{3}x_2 + 5x_3 + \frac{40}{3}x_4 &\leq \frac{275}{3} \\
4x_1 + x_2 + 5x_3 + 3x_4 &\leq \frac{25}{3}x_1 + \frac{5}{3}x_2 + 5x_3 + \frac{40}{3}x_4 \\
z^* &\leq \frac{275}{3}
\end{aligned}$$

Furthermore, by adding constraints 2 and 3, we obtain:

$$\begin{aligned}
4x_1 + x_2 + 5x_3 + 3x_4 \leq 4x_1 + 3x_2 + 6x_3 + 3x_4 &\leq 58 \\
z^* &\leq 58
\end{aligned}$$

We can see that linear combinations of inequality constraints with non-negative multiplier yield valid upper bounds.

We have found a general strategy to calculate an upper bound on the optimal solution of a linear programming problem: linearly combine the constraints with non-negative multiplicative factors. In the example above, we have used the factors:

1. $y_1 = 0, y_2 = 5/3, y_3 = 0$
2. $y_1 = 0, y_2 = 1, y_3 = 1$

In general, any linear combination of the constraints 1, 2 and 3 reads:

$$\begin{aligned} y_1(x_1 - x_2 - x_3 + 3x_4)+ \\ y_2(5x_1 + x_2 + 3x_3 + 8x_4)+ \\ y_3(-x_1 + 2x_2 + 3x_3 - 5x_4) \leq y_1 + 55y_2 + 3y_3 \end{aligned}$$

Which is equivalent to:

$$\begin{aligned} (y_1 + 5y_2 - y_3)x_1+ \\ (-y_1 + y_2 + 2y_3)x_2+ \\ (-y_1 + 3y_2 + 3y_3)x_3+ \\ (3y_1 + 8y_2 - 5y_3)x_4 \leq y_1 + 55y_2 + 3y_3 \end{aligned}$$

In order to use the left side of the inequality as an upper bound on our objective function $z = 4x_1 + x_2 + 5x_3 + 3x_4$, we must have:

$$\begin{cases} y_1 & +5y_2 & -y_3 & \geq 4 \\ -y_1 & +y_2 & +2y_3 & \geq 1 \\ -y_1 & +3y_2 & +3y_3 & \geq 5 \\ 3y_1 & +8y_2 & -5y_3 & \geq 3 \\ y_1, & y_2, & y_3 & \geq 0 \end{cases}$$

In such a case, any feasible solution satisfies the inequality:

$$4x_1 + x_2 + 5x_3 + 3x_4 \leq y_1 + 55y_2 + 3y_3$$

In particular, we can say:

$$z^* \leq y_1 + 55y_2 + 3y_3$$

So we have formulated the so called **dual problem**. Our primal problem was:

$$\begin{array}{llllllll} \max & z & = 4x_1 & +x_2 & +5x_3 & +3x_4 & & \\ \text{s.t.} & & x_1 & -x_2 & -x_3 & +3x_4 & \leq 1 & (1) \\ & & 5x_1 & +x_2 & +3x_3 & +8x_4 & \leq 55 & (2) \\ & & -x_1 & +2x_2 & +3x_3 & -5x_4 & \leq 3 & (3) \\ & & x_1, & x_2, & x_3, & x_4 & \geq 0 & \end{array}$$

And now we try to find an upper bound to its solution by solving our dual problem:

$$
\begin{array}{llllll}
\min & y_1 & +55y_2 & +3y_3 & \\
\text{s.t.} & y_1 & +5y_2 & -y_3 & \geq 4 \\
 & -y_1 & +y_2 & +2y_3 & \geq 1 \\
 & -y_1 & +3y_2 & +3y_3 & \geq 5 \\
 & 3y_1 & +8y_2 & -5y_3 & \geq 3 \\
 & y_1, & y_2, & y_3 & \geq 0
\end{array}
$$

Dual problems

The general form for primal and dual problems of this kind is:

$$
(P) \quad \begin{array}{ll} \max & z = \underline{c}^T \underline{x} \\ \text{s.t.} & A\underline{x} \leq \underline{b} \\ & \underline{x} \geq \underline{0} \end{array} \qquad (D) \quad \begin{array}{ll} \min & w = \underline{b}^T \underline{y} \\ \text{s.t.} & A^T \underline{y} \geq \underline{c} \\ & \underline{y} \geq \underline{0} \end{array}
$$

For standard form, we have:

$$
(P) \quad \begin{array}{ll} \min & z = \underline{c}^T \underline{x} \\ \text{s.t.} & A\underline{x} = \underline{b} \\ & \underline{x} \geq 0 \end{array} \qquad (D) \quad \begin{array}{ll} \max & w = \underline{b}^T \underline{y} \\ \text{s.t.} & A^T \underline{y} \leq \underline{c} \\ & \underline{y} \in \mathbb{R} \end{array}
$$

We can now enunciate the **weak duality theorem**:

Weak duality theorem

Given the primal and dual problems:

$$
(P) \quad \begin{array}{ll} \min & z = \underline{c}^T \underline{x} \\ \text{s.t.} & A\underline{x} \geq \underline{b} \\ & \underline{x} \geq \underline{0} \end{array} \qquad (D) \quad \begin{array}{ll} \max & w = \underline{b}^T \underline{y} \\ \text{s.t.} & A^T \underline{y} \leq \underline{c} \\ & \underline{y} \geq \underline{0} \end{array}
$$

With $X = \{\underline{x} \in \mathbb{R}^n : A\underline{x} \geq \underline{b}, \underline{x} \geq \underline{0}\} \neq \emptyset$ and $Y = \{\underline{y} \in \mathbb{R}^m : A^T\underline{y} \leq \underline{c}, \underline{y} \geq \underline{0}\} \neq \emptyset$. For every feasible solution $\underline{x} \in X$ of (P) and every feasible solution $\underline{y} \in Y$ of (D) we have:

$$
\underline{b}^T \underline{y} \leq \underline{c}^T \underline{x}
$$

As a consequence of this theorem, if $\underline{x}$ is a feasible solution of (P), $\underline{y}$ is a feasible solution of (D) and $\underline{c}^T\underline{x} = \underline{b}^T\underline{y}$, then $\underline{x}$ **is optimal for** (P) **and** $\underline{y}$ **is optimal for** (D).

So we can enunciate the **strong duality theorem**:

> **Strong duality theorem**
>
> If $X = \{\underline{x} \in \mathbb{R}^n : A\underline{x} \geq \underline{b}, \underline{x} \geq \underline{0}\} \neq \emptyset$ and $\min\ \{\underline{c}^T\underline{x} : \underline{x} \in X\}$ is finite, there exist $\underline{x}^* \in X$ and $\underline{y}^* \in Y$ such that:
>
> $$\underline{c}^T\underline{x}^* = \underline{b}^T\underline{y}^*$$

For any pair of primal-dual problems, only four cases can arise among the following:

$(P) \setminus (D)$	**Finite optimal solution**	**Unbounded problem**	**Infeasible problem**
Finite optimal solution	**Yes**, for strong duality	No	No
Unbounded problem	No	No	**Yes**, for weak duality
Infeasible problem	No	**Yes**, for weak duality	**Yes**

3.5.1. Optimality conditions

Given the pair of problems:

$$(P)\quad \begin{array}{ll} \min & z = \underline{c}^T\underline{x} \\ \text{s.t.} & A\underline{x} \geq \underline{b} \\ & \underline{x} \geq \underline{0} \end{array} \qquad (D)\quad \begin{array}{ll} \max & w = \underline{b}^T\underline{y} \\ \text{s.t.} & \underline{y}^T A \leq \underline{c}^T \\ & \underline{y} \geq \underline{0} \end{array}$$

Two feasible solutions $\underline{x}^* \in X$ and $\underline{y}^* \in Y$, with $X = \{\underline{x} \in \mathbb{R}^n : A\underline{x} \geq \underline{b}, \underline{x} \geq \underline{0}\}$ and $Y = \{\underline{y} \in \mathbb{R}^m : \underline{y}^T A \leq \underline{c}^T, \underline{y} \geq \underline{0}\}$ are optimal if and only if $\underline{y}^{*T}\underline{b} = \underline{c}^T\underline{x}^*$.

If x_j and y_i are unknown, this is a single equation in $n + m$ unknowns.

However, since $\underline{y}^{*T}\underline{b} \leq \underline{y}^{*T}A\underline{x}^* \leq \underline{c}^T\underline{x}^*$, we have:

$$\begin{aligned} \underline{y}^{*T}\underline{b} &= \underline{y}^{*T}A\underline{x}^* \\ \underline{y}^{*T}A\underline{x}^* &= \underline{c}^T\underline{x}^* \end{aligned}$$

Therefore:

$$\underline{y}^{*T}(A\underline{x}^* - \underline{b}) = \underline{0}$$
$$(\underline{c}^T - \underline{y}^{*T}A)\underline{x}^* = \underline{0}$$

These are $n + m$ equations in $n + m$ unknowns, and so **necessary and sufficient optimality conditions**.

Complementary slackness conditions

$\underline{x}^* \in X$ and $\underline{y}^* \in Y$ are optimal solutions of, respectively, (P) and (D), if and only if:

$$\begin{aligned} y_i^* \overbrace{(\underline{a}_i^T \underline{x}^* - b_i)}^{s_i} &= 0, \qquad i = 1, \dots, m \\ \underbrace{(c_j^T - \underline{y}^{*T} A_j)}_{s'_j} x_j^* &= 0, \qquad j = 1, \dots, n \end{aligned}$$

Where:

- $\underline{a}_i$ denotes the i-th row of A
- A_j denotes the j-th column of A
- s_i is the slack of the i-th constraint of (P)
- s'_j is the slack of the j-th constraint of (D)
  At optimality, the product of each variable with the corresponding slack variable of the constraint of the relative dual is zero.

3.6. Sensitivity Analysis

Sensitivity analysis is the process used to evaluate the sensitivity of an optimal solution with regards to variations in the model parameters.

Let's start with an example about production planning:

$$\begin{array}{lll} \max & \sum_{j=1}^n p_j x_j & \\ \text{s.t.} & \sum_{j=1}^n a_{ij} x_j \leq b_i & i = 1, \dots, m \\ & x_j \geq 0 & j = 1, \dots, n \end{array}$$

In the above problem:

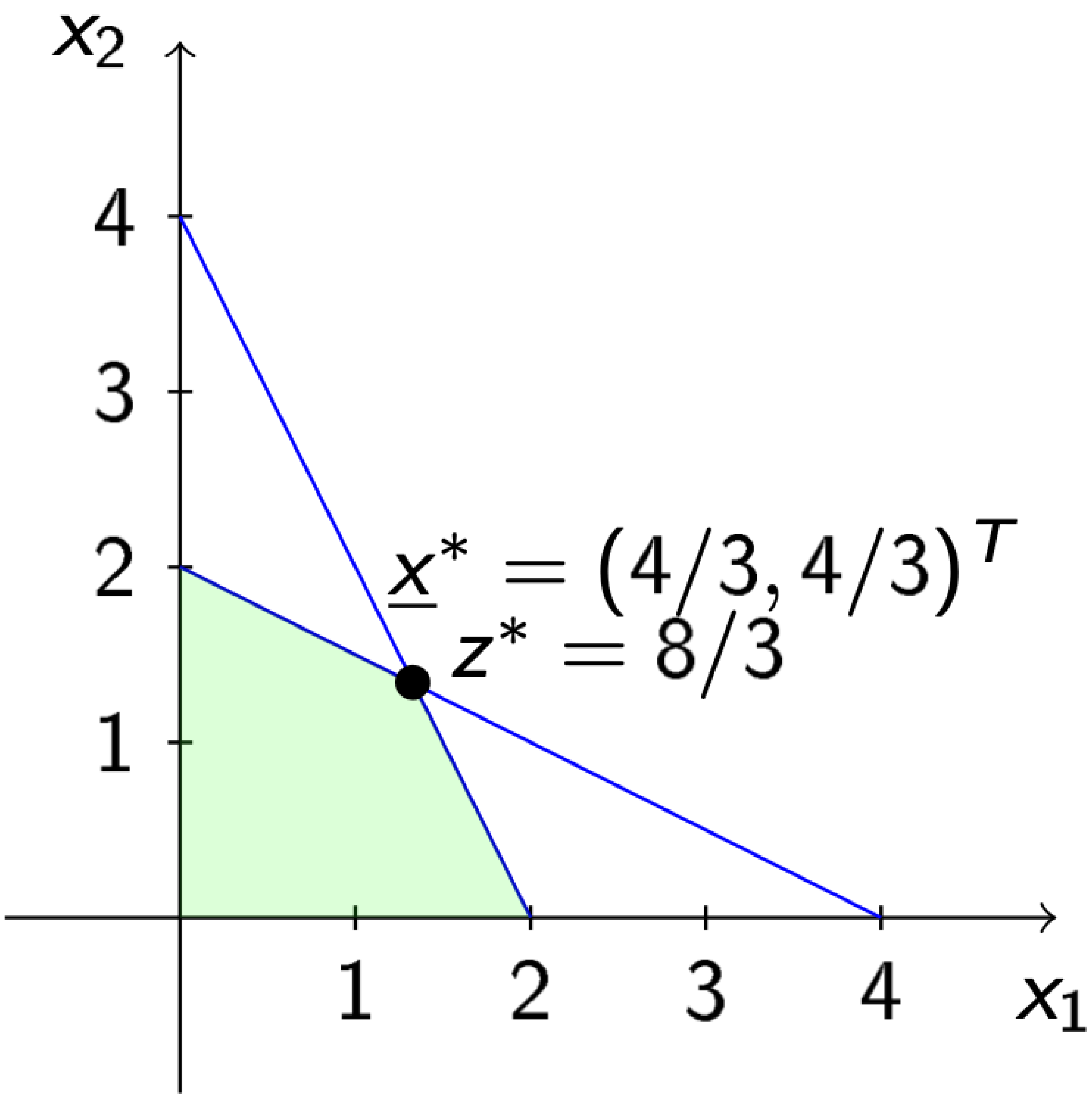
- p_j is the profit of one unit of the j-th product
- b_i is the availability of the i-th resource

3.6.1. Geometric interpretation

Let's try to understand the geometric interpretation of sensitivity analysis taking the following problem into account:

$$\begin{array}{llll} \max & x_1 & +x_2 & \\ \text{s.t.} & \frac{x_1}{2} & +x_2 & \leq 2 \\ & 2x_1 & +x_2 & \leq 4 \\ & x_1, & x_2, & \geq 0 \end{array}$$

We can graph this problem like so:

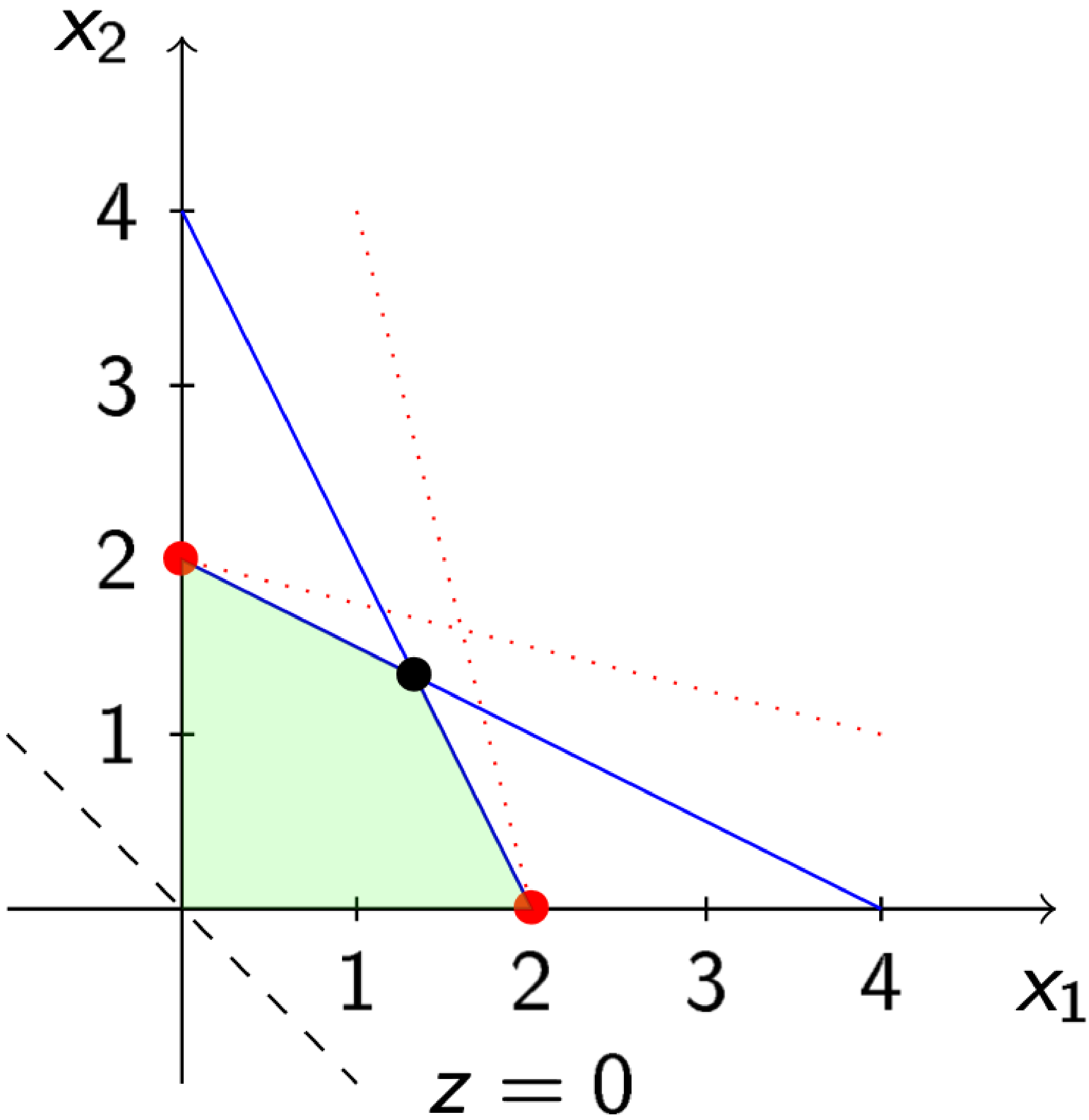


We find that the optimal basic feasible solution is $\underline{x}^* = \left(\frac{4}{3}, \frac{4}{3}\right)^T$, which corresponds to $z^* = \frac{8}{3}$.

Now, if we multiply the variable x_1 by a coefficient c_1, we can see how the graph shifts with regards to the value of c_1:

$$\begin{array}{llll} \max & {\color{red}c_1}x_1 & +x_2 & \\ \text{s.t.} & \frac{x_1}{2} & +x_2 & \leq 2 \\ & 2x_1 & +x_2 & \leq 4 \\ & x_1, & x_2, & \geq 0 \end{array}$$

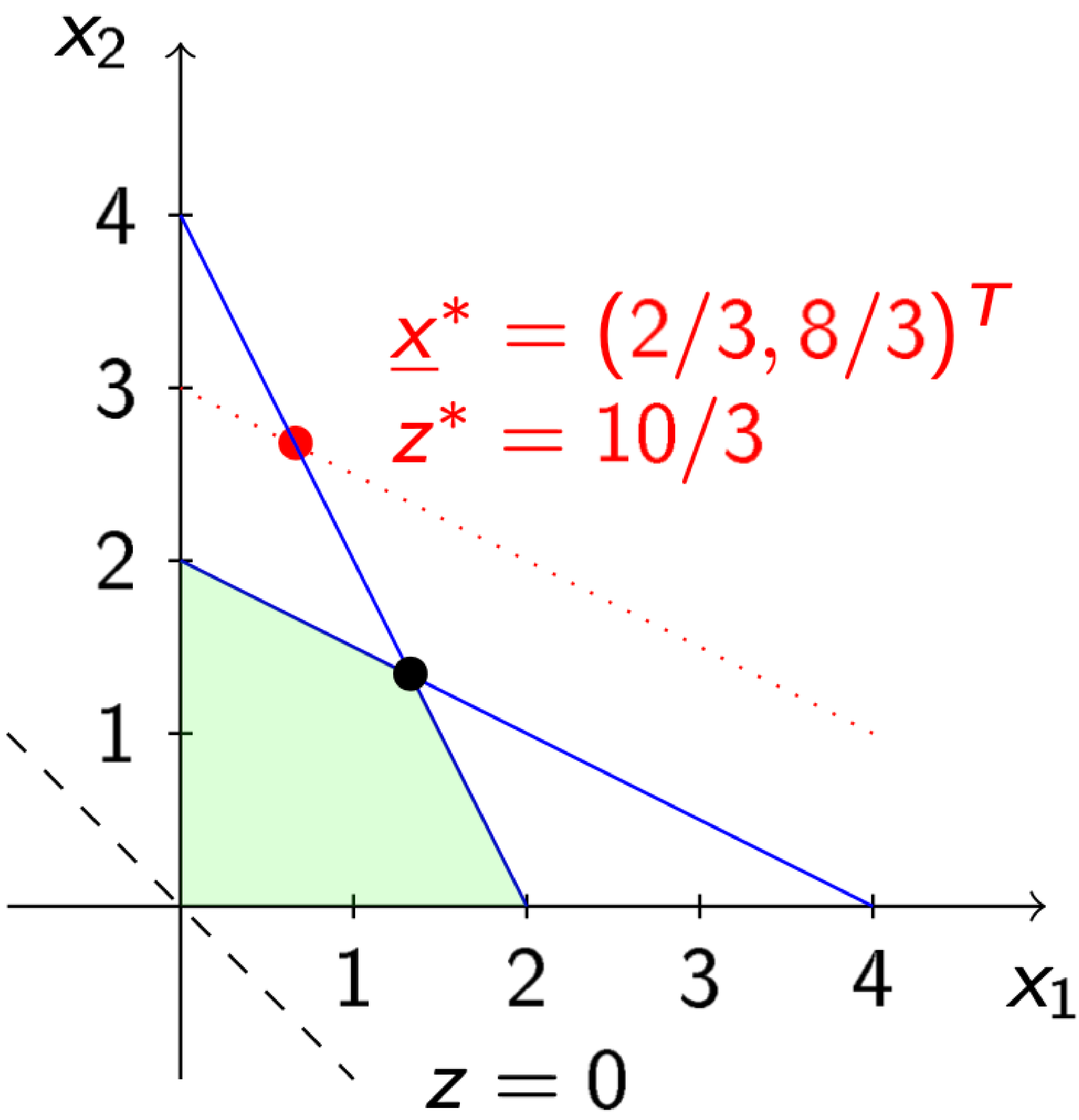
When $\frac{1}{2} \leq c_1 \leq 2$, the graph looks like this:



We can also try modifying the right side of the inequalities:

$$
\begin{array}{llll}
\max & x_1 & +x_2 & \\
\text{s.t.} & \frac{x_1}{2} & +x_2 & \leq 2 \quad {\color{red}+1} \\
 & 2x_1 & +x_2 & \leq 4 \\
 & x_1, & x_2, & \geq 0
\end{array}
$$

We obtain a different optimal solution and graph:



We can define the **shadow price** of the i-th resource as the **maximum price** a company is willing to pay to buy **an additional unit** of the i-th resource.

In the example above, the shadow price of the first resource is:

$$z^* - x_2^* = \frac{10}{3} - \frac{8}{3} = \frac{2}{3}$$

3.6.2. Algebraic form

Sensitivity analysis can be performed algebraically as well.

Given the Linear Programming problem:

$$\begin{aligned} \min \quad & \underline{c}^T \underline{x} \\ \text{s.t.} \quad & A\underline{x} = \underline{b} \\ & \underline{x} > \underline{0} \end{aligned}$$

and an optimal basic feasible solution $\underline{x}^*$ composed of:

$$\begin{aligned} \underline{x}_B^* &= B^{-1}\underline{b} \geq \underline{0} \\ \underline{x}_N^* &= \underline{0} \end{aligned}$$

we want to understand which limits allow the basis B to remain optimal.

We need to respect two conditions:

1. **Feasibility**:

$$B^{-1}\underline{b} \geq \underline{0}$$

2. **Optimality**:

$$\underline{\bar{c}}_N^T = \underline{c}_N^T - \underline{c}_B^T B^{-1} N \geq \underline{0}^T$$

Let's consider the variation of the right-hand side terms first. Let's suppose we have:

$$\underline{b}' := \underline{b} + \delta_k \underline{e}_k, \ 1 \leq k \leq n$$

where $\underline{e}_k$ is the vector which has a single one in the k-th position.

In this case, the basis B with the basic feasible solution:

$$\underline{x}^* = \begin{bmatrix} B^{-1}(\underline{b} + \delta_k \underline{e}_k) \\ \underline{0} \end{bmatrix}$$

remains optimal as long as:

$$B^{-1}(\underline{b} + \delta_k \underline{e}_k) \geq \underline{0}$$

which can be also written as:

$$B^{-1}\underline{b} \geq -\delta_k B^{-1}\underline{e}_k$$

These m inequalities define an interval of variation for δ_k.

Under these conditions, B remains optimal, but the optimal basic feasible solution changes: the objective function value goes from $\underline{c}_B^T B^{-1}\underline{b}$ to $\underline{c}_B^T B^{-1}(\underline{b} + \delta_k \underline{e}_k)$, thus:

$$\Delta z^* = c_B^T B^{-1}(\delta_k \underline{e}_k)$$

Renaming $c_B^T B^{-1}$ to $\underline{y}^{*T}$ (the optimal solution of the dual problem), we obtain the **shadow price** expression:

$$\Delta z^* = \delta_k y_k^*$$

Let's now see what happens if we change the cost coefficients.

Given $\underline{c}' := \underline{c} + \delta_k \underline{e}_k$, a basis B remains optimal as long as:

$$\underline{\overline{c}}'^T_N = \underline{c}'^T_N - \underline{c}'^T_B B^{-1} N \geq \underline{0}$$

In that case, the optimal basic feasible solution does not change:

$$\underline{x}_B^* = B^{-1}\underline{b}$$
$$\underline{x}_N^* = \underline{0}$$

If x_k is a non-basic variable, then the reduced cost of such variable is the maximum decrease of c_k for which B remains optimal. In that case, we don't see a variation in the optimal objective function value z^*.

If x_k is a basic variable instead, we do see a difference in the optimal objective function value, equal to:

$$\Delta z^* = \delta_k x_k^*$$

---

1. For maximisation problems, we check for $\underline{\overline{c}}_N \leq \underline{0}$. ↩